



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/662,087	09/12/2003	Liem Gioi Tran	SVL920030083US1/IBMP029	8862

63056 7590 11/04/2008
MOLLBORN PATENTS, INC.
ATTN: IBM
2840 COLBY DRIVE
BOULDER, CO 80305

EXAMINER

SMITH, GARRETT A

ART UNIT	PAPER NUMBER
----------	--------------

2168

NOTIFICATION DATE	DELIVERY MODE
-------------------	---------------

11/04/2008

ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

MOLLBORN@MOLLBORN.COM
sbailey@mollborn.com

Office Action Summary	Application No.	Applicant(s)	
	10/662,087	TRAN, LIEM GIOI	
	Examiner	Art Unit	
	Garrett Smith	2168	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 24 July 2008.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1,3-8,10-15 and 17-20 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1,3-8,10-15 and 17-20 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This Office action is regarding Applicant's response filed 24 July 2008 to a prior Office action. Claims 1, 3 – 8, 10 – 15 and 17 – 20 are pending.
2. This Office Action is the **Fourth Action, Final Rejection**.

Response to Arguments

35 USC § 103(a)

3. Applicant's arguments (page 5 – 7), filed 24 July 2008, regarding the rejection under 35 USC § 103(a) of claims 1, 3 – 8, 10 – 15 and 17 – 20 have been fully considered but they are not persuasive.

Applicant argues, "No data type is stored for the optimized tree". The Examiner respectfully disagrees. When something is stored in memory or on disk storage in a computer system, a "data type" is always stored because the item (in this case the tree) is its own data type (at minimum).

Applicant further argues, "This overloading is not an overloading of casting functions but simply referring to in Kaneko as an overloading of 'routines'." The Examiner submits that a "routine" is equivalent to a "function" in many programming languages such as C++. A cast function in a basic sense is merely a data type changer function. For example, if there exists an integer J and J equals 1, J can be cast as a floating point number: K = float (J). This results in K = 1.0, newly cast as a float version of J. As such, overloading a casting function (examples int(); float(); long(); etc.) is no

Art Unit: 2168

different from overloading any other function such as operators (+; -, /; *; and so on) or such as user defined functions.

For these reasons, the rejection under 35 USC § 103(a) of claims 1, 3 – 8, 10 – 15 and 17 – 20 is **maintained**.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

5. Claims **1, 3 – 8, 10 – 15 and 17 – 20** are rejected under 35 U.S.C. 103(a) as being unpatentable over Zeller et al (US Patent 5,724,570, patented 3 March 1998) in view of Kaneko et al (US Patent 6,480,833, patented 12 November 2002).

6. In regard to **claim 1**, Zeller et al teaches replacing each of the two or more tokens of the SQL template to form a converted SQL template (*the SQL query is converted by a normalizer, see col 7, lines 1 – 17; also see Fig 3A and 3B as well as col 11, lines 21 – 30; the nested queries are replaced*) and storing the data type of the converted SQL template with the SQL template (*the data type and the can be stored in memory or on hard disk F100, see col 7, lines 1 – 17*).

However, Zeller et al does not teach the explicit use of a cast function. Kaneko et al discusses overloading of casting function and determinations of datatypes of input variables as well as the entire SQL statement. “Overloading a function” is a process by

Art Unit: 2168

which on determination of the number and datatype of variables in calling a function, a particular function is selected to be executed. A determination of datatype must occur in order for the correct function to be selected (*See Figures 9, 14 for selecting of appropriate function based on datatype*). It would have been obvious to a person of ordinary skill in the art to use the method of Zeller et al with the cast function of Kaneko et al because the cast function allows for queries with non-homogeneous data types to be operated on without data type errors.

7. In regard to **claim 3**, Zeller et al further teaches forming a valid SQL statement from the converted SQL template (*from an optimized query tree, equivalent SQL statement can be formed and executed by the executor module F110, see col 7, lines 1 – 17*).

8. In regard to **claim 4**, Zeller et al does not teach the explicit use of a cast function to determine validity of SQL statement. However, as shown by SQL-92 (incorporated by reference), validity of a query can be determined by the results of a cast function.

9. In regard to **claim 5**, Zeller et al further teaches acquiring the data type of the converted SQL statement comprises passing the valid SQL statement through an SQL processor (*the data type is known as the query is acted on, such as BOOLEAN, see col 8, lines 27 – 31; the query is acted on by the optimizer with is equivalent to a SQL processor*).

10. In regard to **claim 6**, Zeller et al further teaches inquiring if a descendent of the converted SQL template has been modified (*the optimizer and normalizer does*

Art Unit: 2168

operations based on rules which will transverse the tree and check for modifications, col 7, lines 1 – 17).

11. In regard to **claim 7**, Zeller et al further teaches if the descendent of the converted SQL template has been modified, re-evaluating an SQL template for the descendent and cascading a modified data type up to ancestors of the converted SQL template (*the optimizer and normalizer does operations based on rules which will transverse the tree and check for modifications, col 7, lines 1 – 17, data types are moved up to the parent query as necessary for computation*).

12. In regard to **claim 8**, Zeller et al teaches replacing each of the two or more tokens of the SQL template to form a converted SQL template (*the SQL query is converted by a normalizer, see col 7, lines 1 – 17; also see Fig 3A and 3B as well as col 11, lines 21 – 30; the nested queries are replaced*) and storing the data type of the converted SQL template with the SQL template (*the data type and the can be stored in memory or on hard disk F100, see col 7, lines 1 – 17*).

However, Zeller et al does not teach the explicit use of a cast function. Kaneko et al discusses overloading of casting function and determinations of datatypes of input variables as well as the entire SQL statement. “Overloading a function” is a process by which on determination of the number and datatype of variables in calling a function, a particular function is selected to be executed. A determination of datatype must occur in order for the correct function to be selected (*See Figures 9, 14 for selecting of appropriate function based on datatype*). It would have been obvious to a person of ordinary skill in the art to use the method of Zeller et al with the cast function of Kaneko

Art Unit: 2168

et al because the cast function allows for queries with non-homogeneous data types to be operated on without data type errors.

13. In regard to **claim 10**, Zeller et al further teaches forming a valid SQL statement from the converted SQL template (*from an optimized query tree, equivalent SQL statement can be formed and executed by the executor module F110, see col 7, lines 1 – 17*).

14. In regard to **claim 11**, Zeller et al does not teach the explicit use of a cast function to determine validity of SQL statement. However, as shown by SQL-92 (incorporated by reference), validity of a query can be determined by the results of a cast function.

15. In regard to **claim 12**, Zeller et al further teaches acquiring the data type of the converted SQL statement comprises passing the valid SQL statement through an SQL processor (*the data type is known as the query is acted on, such as BOOLEAN, see col 8, lines 27 – 31; the query is acted on by the optimizer with is equivalent to a SQL processor*).

16. In regard to **claim 13**, Zeller et al further teaches inquiring if a descendent of the converted SQL template has been modified (*the optimizer and normalizer does operations based on rules which will transverse the tree and check for modifications, col 7, lines 1 – 17*).

17. In regard to **claim 14**, Zeller et al further teaches if the descendent of the converted SQL template has been modified, re-evaluating an SQL template for the descendent and cascading a modified data type up to ancestors of the converted SQL

Art Unit: 2168

template (*the optimizer and normalizer does operations based on rules which will transverse the tree and check for modifications, col 7, lines 1 – 17, data types are moved up to the parent query as necessary for computation*).

18. In regard to **claim 15**, Zeller et al teaches replacing each of the two or more tokens of the SQL template to form a converted SQL template (*the SQL query is converted by a normalizer, see col 7, lines 1 – 17; also see Fig 3A and 3B as well as col 11, lines 21 – 30; the nested queries are replaced*) and storing the data type of the converted SQL template with the SQL template (*the data type and the can be stored in memory or on hard disk F100, see col 7, lines 1 – 17*).

However, Zeller et al does not teach the explicit use of a cast function. Kaneko et al discusses overloading of casting function and determinations of datatypes of input variables as well as the entire SQL statement. “Overloading a function” is a process by which on determination of the number and datatype of variables in calling a function, a particular function is selected to be executed. A determination of datatype must occur in order for the correct function to be selected (*See Figures 9, 14 for selecting of appropriate function based on datatype*). It would have been obvious to a person of ordinary skill in the art to use the method of Zeller et al with the cast function of Kaneko et al because the cast function allows for queries with non-homogeneous data types to be operated on without data type errors.

19. In regard to **claim 17**, Zeller et al further teaches forming a valid SQL statement from the converted SQL template (*from an optimized query tree, equivalent SQL*

Art Unit: 2168

statement can be formed and executed by the executor module F110, see col 7, lines 1 – 17).

20. In regard to **claim 18**, Zeller et al does not teach the explicit use of a cast function to determine validity of SQL statement. However, as shown by SQL-92 (incorporated by reference), validity of a query can be determined by the results of a cast function.

21. In regard to **claim 19**, Zeller et al further teaches acquiring the data type of the converted SQL statement comprises passing the valid SQL statement through an SQL processor (*the data type is known as the query is acted on, such as BOOLEAN, see col 8, lines 27 – 31; the query is acted on by the optimizer with is equivalent to a SQL processor*).

22. In regard to **claim 20**, Zeller et al further teaches if the descendent of the converted SQL template has been modified, re-evaluating an SQL template for the descendent and cascading a modified data type up to ancestors of the converted SQL template (*the optimizer and normalizer does operations based on rules which will transverse the tree and check for modifications, col 7, lines 1 – 17, data types are moved up to the parent query as necessary for computation*).

Conclusion

23. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Garrett Smith whose telephone number is (571)270-1764. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tim T. Vo can be reached on (571) 272-3642. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Art Unit: 2168

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

November 3, 2008

/GS/
Garrett Smith
Patent Examiner
Art Unit 2168

/Tim T. Vo/
Supervisory Patent Examiner, Art Unit 2168